Offline Handwritten Text Recognition on the IAM Database using Deep Neural Networks

Knoop, Aveline
DS6050: Deep Learning, Spring 2024
University of Virginia
Charlottesville, VA, USA
adk8cy@virginia.edu

Maldonado, Wilmer
DS6050: Deep Learning, Spring 2024
University of Virginia
Charlottesville, VA, USA
etc7fq@virginia.edu

Teelucksingh, Victor
DS6050: Deep Learning, Spring 2024
University of Virginia
Charlottesville, VA, USA
vat5jy@virginia.edu

***Abstract*—We investigate training methodology for an offline handwritten text recognition (HWR) model using deep learning approaches. Our research aims to develop accurate models capable of interpreting diverse handwriting styles, with potential applications in accessibility tools for individuals with visual impairments or learning disabilities. Leveraging the IAM Handwritten Dataset, we explore two main deep learning architectures: convolutional neural networks (CNN) and hybrid handwritten text recognition (H2TR) combining CNNs with recurrent neural networks (RNN-LSTM). Using character error rate (CER) and word predictions accuracy (WPA) as evaluation metrics, we performed preliminary experiments with a baseline CNN model. Our work thus far will provide for future experiments with RNNs, hybrid models, and transfer learning strategies as we continue our experimentation. We conclude with potential contributions to the field and avenues for future research, including end-to-end HWR systems, robustness to noise and variability, low-resource recognition, and multilingual capabilities.**

***Index Terms*—computer vision, handwriting recognition, OCR, text recognition**

I. Introduction and Background

In this project, we will explore a deep learning approach to offline handwriting recognition research. Offline handwriting recognition models are designed to classify and accurately label handwritten text as words and sentences. The term "offline" refers to the method of data collection, wherein text is manually written by individuals using pen or pencil on paper, rather than through technological means. Ideally, the model should be capable of recognizing not just one person's handwriting, but a diverse range of writing styles.

The capabilities of a highly accurate handwriting recognition model have numerous use cases. For example, banks utilize it for mobile bank app scanning and uploading of bank checks. Similarly, post offices rely on these models to swiftly process handwritten addresses on envelopes, directing mail to the appropriate destinations without human intervention [1]. Accuracy is crucial in such scenarios, as errors could lead to costly and frustrating mistakes.

These task-specific models are trained contextually for their specific tasks. Consequently, our aim is to enhance these models to develop a handwriting recognition system that doesn't require contextual input, but instead possesses broad linguistic knowledge [1]. The goal is to develop the best neural network architecture model trained on handwritten text images to accurately interpret text images as words, irrespective of the context or domain of application. The handwritten text recognition model can be used to develop accessibility tools for individuals with visual impairments or learning disabilities [2]. By converting handwritten notes into digital text, the model makes written information more accessible through assistive technologies like text-to-speech systems. This can help these individuals access content more easily and participate in digital communication more effectively.

II. Literature Review

A. Current State of Knowledge

In March 2024, Global Data published findings projecting the optical character recognition (OCR) market value to reach 11.7 billion in 2023, and their market research cited an expected growth rate of 15.4% from 2023 to 2030 [5]. OCR technology, which includes handwriting recognition models, facilitates the transformation of printed and handwritten documents into machine-readable text. What are the current methodologies, applications, and deployment strategies of computer vision-based neural networks? Our Proposal aims to leverage computer vision and deep learning technologies – accelerated by the growing OCR industry – to investigate machine understanding of human writing.

Handwriting recognition (HWR) has a wide range of applications across various industries and domains: document digitalization (handwritten text, image-only PDFs), pharmaceutical drug prescriptions, and check verification (banking) [6]. HWR poses challenges for models due to unique data. Huge variations in handwriting, noise, cursive script, deviations from straight-line text, and limited labeled training data hinder accuracy. How have these challenges influenced model development? Scholarly research in this field has investigated approaches including but not limited to convolution neural networks (CNNs), recurrent neural networks (RNNs), hybrid approaches (H2TR), and transfer learning.

Top state-of-the-art contributions in the field typically follow an approach utilizing convolutional neural networks (CNNs) in some capacity, although there are some classical machine learning techniques which are still able to provide competitive error rates [10]. Generally, recent publications

train models using CNN architectures or a hybrid approach using both CNN and RNN/LSTM architectures. Widespread throughout the field is transfer learning, or training models for a specific task after initializing weights using often well-known and high-performing pre-trained models (BART, RESNET, AlexNet, etc.). Approaches follow similar training and testing pipelines using the following high-level processes:

1) Develop input data / image digitization
2) Preprocessing of image data including noise removal, binarization, resizing, etc. (discussed in detail below)
3) Boundary detection and segmentation
4) Feature extraction
5) Classification and recognition
6) Post-processing and output results

Traditionally the architecture for a CNN model for handwritten character recognition consists of convolutional layers for extracting features and fully connected layers followed by a softmax layer for classification. The convolutional layers extract features by creating feature maps. The output of convolutional layers is typically activated using ReLU to introduce non-linearity. In between convolutional layers we usually see a pooling layer to reduce computation in the network and increase test performance. Max pooling is a common method that generally outperforms other pooling methods like min pooling or average pooling. A fully connected layer learns the non-linear combination of the features and classifies or predicts the output (followed by a softmax layer for classification problems or a regression layer for regression problems). We initialize weights and biases using forward pass and train the network using backpropagation to minimize the chosen error function [11].

We can expand upon this traditional structure by also including the recurrent neural network (RNN) which can potentially lead to improvements because it can process larger input though it has lesser computational power. The hybrid method trains the dataset consecutively with CNN and RNN, and the connectionist temporal classification (CTC) network is fitted along with RNN. We use a similar structure of CNN layers with ReLU activation and max pooling, but each time step consists of feature sequence that is applied to the RNN, which is implemented using a Long Short-Term Memory (LSTM) network since it can transfer the data through a longer range and has superior training characteristics compared to Vanilla RNN. The motivation for this architecture is the potential for time savings, faster processing, and reduced probability for error [12]. This architecture fits into the same high-level workflow activities noted above.

Across the current state of the literature, image preprocessing is a common step in the overall Handwriting Recognition (HWR) workflow (and, more generally, in the Text Recognition workflow) and is used to improve both training and test performance. There are some general preprocessing steps that are common throughout, i.e., image scaling, binarization, noise removal, and segmentation (at the word-level or character-level depending on approach, although character-level generally provides better results with faster training times). However, there is a lack of consensus on the specific methods used to accomplish preprocessing tasks, which may also vary depending on which methods are most appropriate for the data set in question. We see multiple methods used to accomplish similar goals during preprocessing. For example, to address the issue of images of varying size and shape, we see use of image resizing, image padding with whitespace to meet maximum width and height present, and/or normalization although generally, deep learning models often train faster and in a more effective manner when the input image is smaller than a certain dimension [4] [13] [14].

There are also many proposed methods to accomplish segmentation, one of the most crucial steps that can significantly improve the accuracy of HTR models [14]. In particular, threshold methods, region-based methods, edge-based methods, watershed-based methods, and clustering-based methods are used for segmenting at the block, word, character or line level. Generally implementation of segmentation involves transfer-learning from larger pretrained models with parameter tuning to fit the HWR data set, and models used vary widely depending on available models at time of writing and the origin of the data set to be analyzed (English, Arabic, Chinese, etc.). Preprocessing remains a rapidly changing and fiercely debated topic in Handwriting Recognition using deep learning.

B. Related Works

In this investigation we will be basing our neural network architecture on two deep learning modeling techniques for handwritten text recognition systems, convolution neural network (CNN) and hybrid handwritten text recognition (H2TR) model.

H2TR was proposed by R. Geetha, T. Thilagam and T. Padmavathy as a method to leverage the salient features of convolution neural network (CNN) and recurrent neural network (RNN) with long-short-term memory network (LSTM). The H2TR model first uses CNN to extract features from handwritten images. Then, with a sequence-to-sequence (Seq2Seq) approach the extracted features are implemented in a RNN-LSTM network for encoding the visual features and decoding the sequence of letters of the handwritten text images. In word accuracy on word recognition datasets, H2TR (CNN-RNN) consistently outperformed traditional CNN architectures. H2TR model and CNN model, tested on RIMES Dataset, had a word accuracy of 98.14 percent and 96.31 respectively. H2TR model and CNN model, tested on IAM Handwriting Dataset, had a word accuracy of 95.20 percent and 94.17 respectively. [3].

We will leverage these related works to identify the deep learning model architectures that yield the highest test accuracy on the IAM handwritten dataset. Furthermore, we will investigate how the specific characteristics of each model architecture influence their predictive performance in offline handwritten recognition tasks.

C. Methodological Issues

Model development for HWR presents several challenges. Cursive handwriting poses many issues due to the 'blending' of characters. Unique features such as symbols, abbreviations, diverse handwriting styles, and the tendency of handwritten script to not appear along straight lines requires robust algorithms capable of handling massive variability. Noise in handwritten documents, stemming from ink smudges and pen pressure differences can significantly degrade accuracy. Efforts to mitigate these challenges include noise reduction, as highlighted in a study by H. Pham et al. and their proposal for two consecutive stages of word segmentation and word recognition [7]. After segmenting documents into entire word-forms, their best model returned an average precision of 89.1%, greatly outperforming industry baselines EAST and CRAFT (returning 38.9% and 12.8% average precision, respectively). H. Pham et al.'s exploration into a custom and tuned character model also, as expected, resulted in massive under-performance. By opting to apply word segmentation, H. Pham et al.'s work addressed the issues associated with character noise, and their work supports word recognition over individual character recognition. This method is limited to language, however, and digit recognition remains subject to misclassification with the slightest perturbations [8].

Another major obstacle is the scarcity of labeled learning data, particularly for non-English alphabets, which are essential for training effective neural networks. The lack of diverse and comprehensive datasets limits the applicability of models across different languages and unlabeled documents. Research conducted by Paquet et al. proposes a two-step process of recognition, where an OCR model classifies character sequences, followed by a language model to derive probabilistic meaning from the resulting sequences. Paquet et al. applied a multilingual system evaluation using an English and French case study and found that where languages are of the same origin, "unifying the character set reduces the system complexity and increases the number of training examples per character classes that are shared between the languages," [9]. This proposal addresses the challenge of labeled data across languages by using sub-lexical units, or multigrams of language similarity. Their innovative methodology offers a solution to the major shortcomings of multilingual models, and progress in this area will extend to HWR as advancements continue.

D. Proposed Contributions

Our objective is to develop an accurate model capable of interpreting diverse handwriting styles without relying on the contextual constraints typically found in task-specific applications like bank check scanning or postal address text recognition systems [1]. This approach could allow the handwritten recognition system to be used to enhance accessibility tools for individuals with visual impairments or learning disabilities [2]. Based on the findings in the related work, we plan to investigate two deep learning modeling techniques for our offline handwriting recognition system: convolution neural network (CNN) and hybrid handwritten text recognition (H2TR), which combines CNN and RNN-LSTM [3]. Given the diversity in handwriting styles and the need for large amounts of data, we will utilize the IAM Handwritten Dataset for model training, validation, and testing. Despite the challenges posed by preprocessing requirements of the IAM Handwritten Dataset and the need to capture real-world handwriting diversity, we aim to contribute valuable insights to offline handwriting recognition through careful model tuning and leveraging existing literature on deep learning methods.

III. Method

A. Dataset

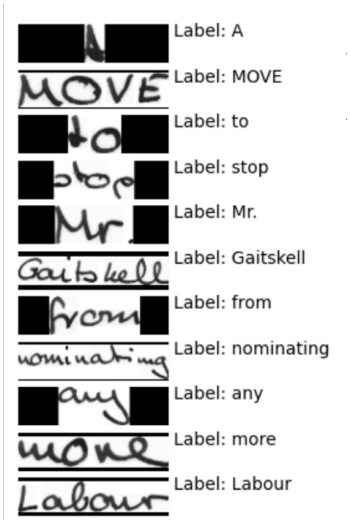

Fig. 1. IAM Dataset sample.

The IAM Handwriting Dataset contains handwritten English text. Images are of PNG file type with a resolution of 300 dpi, with 256 gray levels. The database includes: 657 writers, 1,539 pages of scanned text, 5,685 isolated and labeled sentences, 13,353 isolated and labeled text lines, and 115,320 isolated and labeled words. There are 76 unique characters in the dataset, and they include upper and lowercase letters, punctuation, and digits. The dataset is approximately 5GB.

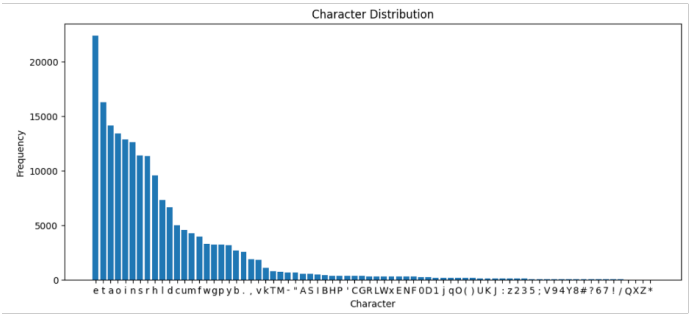

Fig. 2. IAM Dataset character frequency distribution.

The IAM Handwriting Database is commonly used for training and evaluation of offline handwriting recognition models due to its diversity, depth, and quality. Deep learning models require a large amount of data to be sufficiently trained, and this dataset has over 115,000 word instances [4]. IAM's word length distribution follows a distribution consistent with that expected of an English corpus. The high frequency of labels with a length of 1 is the dataset's only discrepancy, and this occurs because, unlike other English corpora, IAM includes digits and punctuation.

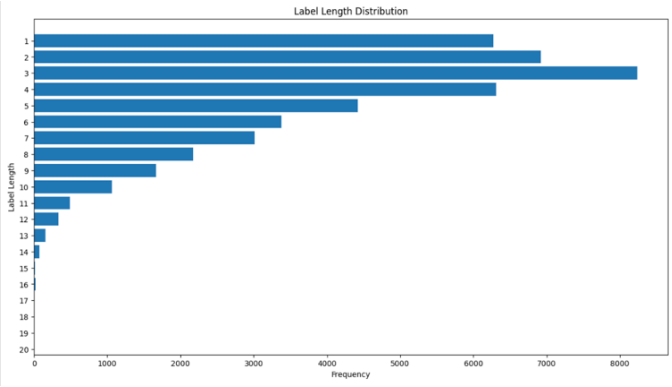

Fig. 3. IAM Dataset label length frequency distribution.

Data can be accessed via Kaggle or from direct source, IAM Handwriting Database 3.0. URL links to access IAM Handwriting Dataset:

- https://www.kaggle.com/datasets/nibinv23/iam-handwriting-word-database/data?select=iam_words
- https://fki.tic.heia-fr.ch/databases/iam-handwriting-database

B. Approach

The IAM handwritten dataset will be split into a training (80%), validation (10%) and test set (10%). We plan to explore different preprocessing techniques to improve quality of data such as data augmentation, image scaling, and picture enhancement. Batching and buffered prefetching techniques will also be used to minimize memory workload and speed up data retrieval.

Once data is configured for performance and preprocessed, we plan to develop two different deep learning models. One model will have a CNN only architecture. The second model will have hybrid handwritten text recognition (H2TR).

The two models will be trained with training set, and parameter tuned to minimize validation loss. Different learning rates, optimizers, activation functions, and number of layers will be experimented to achieve optimal performance for each of the models.

The tuned models will be evaluated on test set by comparing each model's word accuracy. Finally, the results will be compared to existing literature to assess the alignment of our model with research expectations, and to explore how the architecture of each model may explain any variations or consistencies in the findings.

IV. Experiments

For experimentation, we tested two different handwriting recognition (HWR) models. One model uses only Convolutional Neural Network (CNN) architecture. This CNN architecture acts as our baseline CNN model. The second model adds two Bidirectional Long Short Term Memory (LSTM) layers to the baseline model before output layer to create a hybrid CNN-RNN neural network for handwritten text Recognition. Both models were trained and tuned to minimize the Connectionist Temporal Classification (CTC) loss function on validation set, a function commonly used to calculate loss in neural network handwriting recognition tasks.

A. CNN only baseline model

The CNN only model takes an image input, using distortion-free image resizing to preserve aspect ratios along with its corresponding label inputs. Our model architecture utilizes two convolutional layers with ReLu activation for feature extraction. A max pooling layer follows each convolutional layer to reduce the dimensionality and introduce translation invariance. Next, a reshaping layer prepares the outputs for subsequent layers. A dense layer provides additional feature extraction, and a dropout layer prevents overfitting. Another dense layer, equal to the number of unique characters plus two (accounting for a padding token and blank characters), uses the softmax activation function to predict the probability distribution over the characters for each timestep in the sequence. In a final layer, our architecture integrates the CTC loss function, which is often used for sequence recognition use cases. CTC loss trains our model to align inputs with target (label) sequences and handle variable-length outputs by incorporating logic for blank labels and label repetitions. We have compiled our model using the Adam optimizer, a learning rate of 0.001, a batch size of 32, and 50 training epochs. This model has 56,783 trainable parameters and zero non-trainable parameters. With two GPU T4s training duration was approximately 1 min per epoch.

B. CNN-RNN(LSTM) model

The hybrid CNN-RNN(LSTM) model uses almost the same architecture as the baseline CNN model described above except it adds two Bidirectional LSTM layers before the output dense layer. These RNN layers process the input sequence forward and backward and combines the output allowing for better context utilization in final prediction sequence. Due to these added features by the CNN-RNN architecture, we hypothesize this will result in better overall classification performance. This CNN-RNN model will also trained and tuned to minimize CTC loss on validation set. Model training utilized Adam optimizer, a learning rate of 0.001, a batch size of 32, and 50 training epochs. This hybrid model has 423,823 trainable parameters and zero non-trainable parameters. Due to the increased number of trainable parameters increased epochs to 80 for training. With two GPU T4s training duration was approximately 2 min per epoch.

Handwritten Text Recognition performance will be measured by Character Error Rate (CER), overall letter accuracy, and word prediction accuracy (WPA) on unseen data (test set).

CER is calculated by dividing the number of edits required to make prediction match true labels over the total number of characters in the true label. The CER per word, on average, in the test set was recorded as the model's final score. Letter accuracy is calculated by dividing the total number of characters predicted correctly in a dataset by the total number of characters in dataset. Lastly, word prediction accuracy (WPA) is calculated by total number of word predictions that matched true labels over the total number of words in dataset.

C. Results

Training and Validation Set loss for the CNN baseline model after 50 epochs was 7.85 and 7.76 respectively, with an initial loss of 18.70 at epoch 1. Training and Validation Set loss for the CNN-RNN(LSTM) hybrid model after 80 epochs was 1.90 and 2.74 respectively, with an initial loss of 15.45 at epoch 1. CNN baseline model converged at around 40 epochs. Although training per epoch was longer in hybrid CNN-RNN model, it converged slightly faster at around 30 epochs.

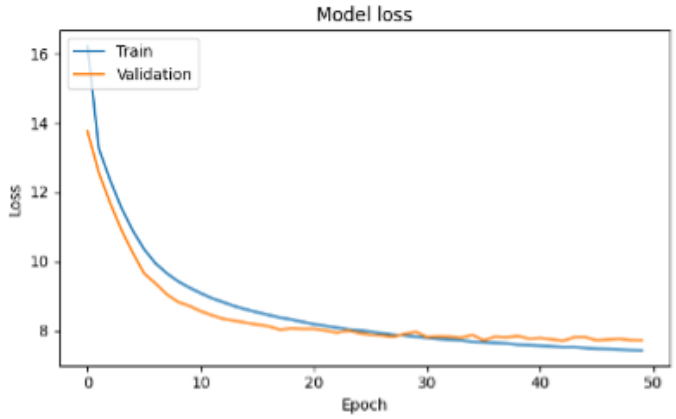

Fig. 4. Baseline model training and validation loss.

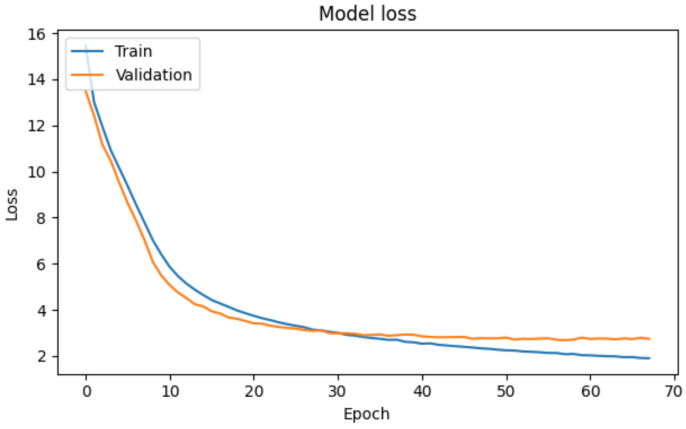

Fig. 5. CNN-RNN (LSTM) model training and validation loss.

Although training and validation results clearly showed the hybrid CNN-RNN model as the better performing model, we further tested both models on unseen data, the test set, and measured more common character and word recognition metrics to gauge overall model performance. The three metrics were, Character Error Rate (CER), overall letter accuracy, and word accuracy.

The CNN only baseline model and CNN-RNN (LSTM) scored a CER of 52.27 percent and 14.60 percent. The baseline CNN model received a test set letter accuracy of 46.04 percent and the CNN-RNN (LSTM) model scored 84.20 percent. Lastly, word prediction accuracy (WPA) was 17.02 percent for the CNN baseline model and 65.15 percent for the CNN-RNN (LSTM) model.

D. Conclusions

For each metric—CER, WPA, and letter accuracy—the hybrid architecture with CNN and Bidirectional LSTM layers outperformed the baseline CNN model. This is evidence of how the RNN layers provide improved contextualized features for enhanced character and word recognition in offline handwritten text. Bidirectional LSTMs contextualize by capturing both past and future context for each time step in a sequence. The current literature supports our conclusion suggesting that LSTM layers provide improved performance for this task.

Additionally, the longer training time per epoch in the LSTM model (by 1 minute compared to the CNN baseline model) is expected due to the approximately 370,000 more trainable parameters than the CNN baseline model. Despite this, its superior performance leads to a quicker convergence during training.

Another notable observation is that when comparing the CNN-RNN predicted results to the actual label, we noticed many words with only 1 or 2 missed letters. To address this issue, one potential solution is to map predictions to a dictionary of words, matching them with the corresponding word in the dictionary with the lowest edit distance between them.

Overall, the CNN-RNN/LSTM model should be the preferred model over the CNN baseline model. With a CER of 14 percent and a letter accuracy of 84 percent, this model demonstrates its suitability for offline handwritten text recognition.

V. Member Contribution

- Wilmer Maldonado: Research, method planning, model evaluation, assist in write-up
- Victor Teelucksingh: Research, method planning, code troubleshooting, assist in write-up
- Aveline Knoop: Research, abstract, lit review, EDA, code review

VI. Future Study

At present, researchers are refining HWR models by classifying noise as distinct artifacts, capturing words as features via segmentation, and extending recognition capabilities across

language systems. Neural networks offer the potential to explore ancient texts, and, therefore, opening new avenues for inquiry into historical and archaeological study. As advancements in deep learning continue, the future of HWR lends itself to greater accuracy, efficiency, and applications across domains. Potential areas of future research:

- End-to-End Handwriting Recognition Systems: Creating a standardized pipeline for handwriting digitization that streamlines both OCR and language models.
- Robustness to Noise and Variability: Models that can handle smudges, uneven strokes, varying writing styles, and writing that does not follow straight line patterns.
- Low-Resource Handwriting Recognition: Models developed with smaller amounts of data for lack of labeled datasets. Semi-supervised or unsupervised learning could be made possible by natural language processes incorporating contextual information.
- Multilingual Handwriting Recognition: Relies on both the development of effective HWR models and multilingual language models (independent of handwriting and faces its own challenges).

REFERENCES

[1] U.-V. Marti and H. Bunke, "A full English sentence database for off-line handwriting recognition," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR '99 (Cat. No.PR00318)*, Bangalore, India, 1999, pp. 705-708, doi: 10.1109/ICDAR.1999.791885.

[2] V. Vilasini, B. Banu Rekha, V. Sandeep, and V. Charan Venkatesh, "Deep Learning Techniques to Detect Learning Disabilities Among children using Handwriting," in *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, 2022, pp. 1710-1717, doi: 10.1109/ICICICT54557.2022.9917890.

[3] R. Geetha, T. Thilagam, and T. Padmavathy, "Effective offline handwritten text recognition model based on a sequence-to-sequence approach with CNN–RNN networks," *Neural Computing and Applications*, vol. 33, no. 17, pp. 10923-10934, Sep. 2021, doi: 10.1007/s00521-020-05556-5.

[4] B. Balci, D. Saadati, and D. Shiferaw, "Handwritten Text Recognition using Deep Learning," in CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:2602287.

[5] Global Data, "Optical Character Recognition Market Size, Trends and Analysis by IT Infrastructure, End-use, Vertical, Region and Segment Forecast, 2023-2030," *ReportLinker Consulting*, 2024. [Online]. Available: https://www.reportlinker.com/p06468815/Optical-Character-Recognition-Market-Size-Trends-and-Analysis-by-IT-Infrastructure-End-use-Vertical-Region-and-Segment-Forecast.html.

[6] A. Matcha, "Handwriting Recognition using Machine Learning," *Nano Net Technologies Inc*, Sep. 2023. [Online]. Available: https://nanonets.com/blog/handwritten-character-recognition.

[7] H. Pham et al., "Robust Handwriting Recognition with Limited and Noisy Data," in *17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Dortmund, Germany, 2020, pp. 301-306, doi: 10.1109/ICFHR2020.2020.00062.

[8] M. Bethge et al., "Towards the first adversarially robust neural network model on MNIST," in *arXiv preprint arXiv:1805.09190*, May 2018.

[9] T. Paquet, Y. Soullard, W. Swaileh, "A Unified Multilingual Handwriting Recognition System using multigrams sub-lexical units," in *Pattern Recognition Letters*, Volume 121, 2019, pp. 68-76, ISSN 0167-8655, doi: 10.1016/j.patrec.2018.07.027.

[10] A. Baldominos, Y. Saez, and P. Isasi, "A Survey of Handwritten Character Recognition with MNIST and EMNIST," in *Applied Sciences*, vol. 9, no. 15, p. 3169, Aug. 2019, doi: https://doi.org/10.3390/app9153169.

[11] M. B. Bora, D. Daimary, K. Amitab, and D. Kandar, "Handwritten Character Recognition from Images using CNN-ECOC," in *Procedia Computer Science*, vol. 167, pp. 2403–2409, 2020, doi: https://doi.org/10.1016/j.procs.2020.03.293.

[12] G. R. Hemanth, M. Jayasree, S. Keerthi Venii, P. Akshaya, and R. Saranya, "CNN-RNN Based Handwritten Text Recognition," vol. 12, no. 1, pp. 2457–2463, Oct. 2021.

[13] A. Kumar and P. B. Pati, "Offline HWR Accuracy Enhancement with Image Enhancement and Deep Learning Techniques," in *Procedia Computer Science*, vol. 218, pp. 35–44, 2023, doi: https://doi.org/10.1016/j.procs.2022.12.399.

[14] W. AlKendi, F. Gechter, L. Heyberger, and C. Guyeux, "Advancements and Challenges in Handwritten Text Recognition: A Comprehensive Survey," in *Journal of Imaging*, vol. 10, no. 1, p. 18, Jan. 2024, doi: https://doi.org/10.3390/jimaging10010018.